DYNAMIC STRUCTURAL ANALYSES USING BOUNDARY ELEMENT TECHNIQUES

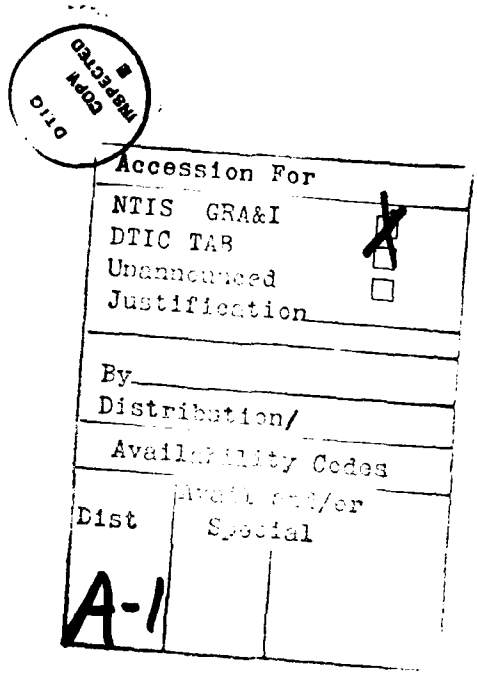
THESIS

Matthew C. Phillips
Captain, USAF

AFIT/GAE/ENY/89D-28

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/89D-28



DYNAMIC STRUCTURAL ANALYSES USING BOUNDARY ELEMENT TECHNIQUES

THESIS

Matthew C. Phillips
Captain, USAF

AFIT/GAE/ENY/89D-28

Approved for public release; distribution unlimited

AFIT/GAE/ENY/89D-28

DYNAMIC STRUCTURAL ANALYSES USING BOUNDARY ELEMENT TECHNIQUES

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Aeronautical Engineering

Matthew C. Phillips, B.S.

Captain, USAF

June 1990

Acknowledgements

I would like to express my sincere thanks to the few individuals who contributed to this effort. First, I must thank my advisor, Major Lanson J. Hudson, for providing the initial concept along with invaluable technical and programing assistance. In addition, a word of thanks to Captain Frank Smith for giving needed encouragement when things looked hopeless. I would also like to acknowledge my Father, Charles L. Phillips, for his wisdom and personal counsel. Finally, a loving thank-you to my wife, Lisa, for her unconditional, loyal and loving support during all of my foolish endeavors.

Table of Contents

List of Figures

List of Symbols

A = cross sectional area

C = spatial density function

E = modulus of elasticity

e_{ij} = strain tensor

$\hat{e}_j$ = unit vector

F_i = body force vector

f = fictitious density function

G = shear modulus

h = height

I = moment of inertia

L = length

m = mass density

n_j = direction cosine

r = distance between two points

t = thickness

t_i = surface traction vector

u_j = displacement vector

$\hat{u}_i$ = arbitrary function

$\{v\}$ = state vector

w = circular frequency

α = eigenvalue exponential coefficient

β_b = phase angle for bending

β_{ce} = phase angle for compression - extension

δ_{ij} = Kronecker delta

$\delta(x,x_o)$ = Dirac delta function

ϵ = spatial eigenvalue

λ = Lame's constant

σ_{ij} = stress tensor

ν = poisson's ratio

Abstract

The purpose of this ~~investigation~~ is to demonstrate the use of boundary element techniques for the dynamic analyses of geometrically repetitive structures using the traveling wave approach. A formulation of the boundary element method (BEM) for 2-D isotropic materials is developed. The BEM formulation is then used to calculate the mass and stiffness matrices of one bay of a baseline structure. From the mass and stiffness matrices a transfer matrix is developed for the bay. Using traveling wave theory, the transfer matrix is then used to identify the dynamical characteristics of a multiple bay structure. Results are compared to continuum theory.

DYNAMIC STRUCTURAL ANALYSES USING BOUNDARY ELEMENT METHODS

I. Introduction

1.1 Background

The advent of high speed computers has led to a revolution in structural analysis methods. Complex numerical algorithms as found in finite element codes are now routinely used in the analysis of everything from coat hangers to spacecraft. As structures of interest become larger and more complex, more accurate and efficient algorithms are developed. Recently, there has been a great deal of interest in the use of boundary element methods (BEM) for structural analysis. BEM offers the advantage of reducing the computational size of the problem compared with traditional finite element methods (FEM). The application of boundary element techniques is particularly attractive in analyzing large repetitive truss-like structures, such as those being proposed for orbiting space platforms. A typical large space structure will be sensitive to wave propagation from on-board disturbances such as gyros, actuators, docking procedures, etc. This thesis introduces the use of boundary element theory in developing the wave propagation transfer matrix for two-dimensional periodic structures.

1.2 Boundary Element Methods

Boundary element theory is used in many disciplines and has proved to be an efficient and elegant method for solving many numerically

intensive problems. The boundary element method, just like the finite element method, is based on the approximate solution of an equation or set of equations describing a physical problem. Unlike FEM however, BEM utilize functions that identically satisfy the governing equations and only approximately satisfy the boundary conditions. In addition, only the boundary of the given problem needs to be discretized when using the BEM. This greatly reduces the modeling effort and results in smaller matrices, although the matrices are often fully populated. A further reduction in problem size is accomplished by combining a traveling wave approach for periodic structures with BEM.

1.3 Wave Propagation

The study of wave propagation has been pursued throughout a wide range of disciplines including solid state physics, fluids, power transmission, etc. Wave propagation theory has also been shown to be a useful tool in the area of dynamic structural analysis. Cremer and Lielich studied flexural motion in periodic beams (1). In 1964 Heckl defined the notion of propagation coefficients in periodically supported, undamped grillages (2:1335-1343). More recently, von Flotow employed the use of wave propagation theory in developing a transfer matrix method for analyzing periodic structures (3:509-519). The transfer matrix method requires that only a single cell of the truss structure be analyzed. In a related effort, Signerolli combined the use of transfer matrices with FEM in analyzing a two-dimensional periodic truss (4).

1.4 Overview

This thesis demonstrates the use of BEM to develop the transfer matrix for a two-dimensional periodic truss. Chapter 2 begins with the development of the boundary element equation for a two-dimensional isotropic material. The boundary element equation is then used to develop the transfer matrix for a simple periodic structure in chapter 3. In Chapter 4, the transfer matrix is used in determining the wave propagation behavior of a baseline beam structure. The results are then compared to equivalent continuum models. Chapter 5 discusses the conclusions and recommendations.

II. The Boundary Element Equation

2.1 The Elastic Dynamic Equation

The governing equation for the behavior of a deformable and continuous body can be written in index notation as

$$\frac{\partial \sigma_{ij}}{\partial x_j} + F_i = m\ddot{u}_i \tag{1}$$

where ∂ is the partial derivative operator, σ_{ij} is the stress tensor, F_i is the body force vector, m is the mass density, and $\ddot{u}_i$ is acceleration. Another useful relationship is the stress-strain equation.

$$\sigma_{ij} = 2Ge_{ij} + \lambda\delta_{ij}e_m \tag{2}$$

Where G is the shear modulus, e_{ij} is the strain tensor, δ_{ij} is the Kronecker delta, and λ is Lame's constant. For linear isotropic material properties, strain can be expressed as

$$e_{ij} = \tfrac{1}{2}(u_{i,j} + u_{j,i}) \tag{3}$$

where u_j is the displacement vector. Eqs. (2) and (3) can be combined to give stress in terms of displacement.

$$\sigma_{ij} = G(u_{i,j} + u_{j,i}) + \lambda\delta_{ij}u_{k,k} \tag{4}$$

Eqs. (1), (2) and (3) represent 15 different equations in 15 unknowns; 6 σ's, 6 e's and 3 u's. When combined, these three equations fully describe the behavior of an elastic domain and will be used to develop the boundary element equation (5,210-211).

To formulate the solution of the elasticity equations, Eq. (1) will be multiplied by an arbitrary function, $\hat{u}_i$, and integrated throughout the domain. Body forces can be neglected.

$$\int_{\Omega} \hat{u}_i \left(\frac{\partial \sigma_{ij}}{\partial x_j} - m\ddot{u}_i \right) d\Omega = 0 \tag{5}$$

This equation is integrated by parts until all of the differentials are on the arbitrary function, $\hat{u}_i$, rather than the unknown displacements. Use the identity

$$\frac{\partial (\hat{u}_i \sigma_{ij})}{\partial x_i} = \hat{u}_{i,j}\sigma_{ij} + \hat{u}_i \sigma_{ij,j} \tag{6}$$

to integrate the first term by parts

$$\int_{\Omega} \left\{ \frac{\partial (\hat{u}_i \sigma_{ij})}{\partial x_i} - \hat{u}_{i,j}\sigma_{ij} - \hat{u}_i m\ddot{u}_i \right\} d\Omega = 0 \tag{7}$$

Using the Guass-Divergence theorem, the volume integral of the first term may be written as a surface integral.

$$\int_\Gamma n_j(\hat{u}_i\sigma_{ij})\ d\Gamma \quad - \int_\Omega (\hat{u}_{i,j}\sigma_{ij} + \hat{u}_i m\ddot{u}_i)\ d\Omega = 0 \tag{8}$$

were n_j is the direction cosine of the outward unit normal vector. Now substitute Eq. 4 under the second integral.

$$\int_\Gamma n_j(\hat{u}_i\sigma_{ij})\ d\Gamma \quad - \int_\Omega \{\hat{u}_{i,j}[G(u_{i,j} + u_{j,i}) + \lambda\delta_{ij}u_{k,k}] + \hat{u}_i m\ddot{u}_i\}\ d\Omega = 0 \tag{9}$$

or

$$\int_\Gamma n_j(\hat{u}_i\sigma_{ij})\ d\Gamma \quad - \int_\Omega \hat{u}_{i,j}Gu_{i,j}\ d\Omega - \int_\Omega \hat{u}_{i,j}Gu_{j,i}\ d\Omega$$

$$- \int_\Omega \hat{u}_{i,i}\lambda u_{k,k}\ d\Omega - \int_\Omega \hat{u}_i m\ddot{u}_i\ d\Omega = 0 \tag{9a}$$

Next, utilize the product rule for differentiation, i.e

udv = d(uv) - vdu

$$\int_\Gamma n_j(\hat{u}_i\sigma_{ij})\ d\Gamma \quad - \int_\Omega (\hat{u}_{i,j}Gu_i)_{,j}\ d\Omega + \int_\Omega (\hat{u}_{i,j}G)_{,j}u_i\ d\Omega$$

$$- \int_\Omega (\hat{u}_{i,j}Gu_i)_{,i}\ d\Omega + \int_\Omega (\hat{u}_{i,j}G)_{,i}u_j\ d\Omega$$

$$- \int_\Omega (\hat{u}_{i,i}\lambda u_k)_{,k}\ d\Omega + \int_\Omega (\hat{u}_{i,i}\lambda)_{,k}u_k\ d\Omega - \int_\Omega \hat{u}_i m\ddot{u}_i\ d\Omega = 0 \tag{10}$$

Now the volume integrals of gradients may be replaced by surface integrals, using the Guass-Divergence Theorem.

$$\int_\Gamma n_j(\hat{u}_i\sigma_{ij})\,d\Gamma - \int_\Gamma n_j(\hat{u}_{i,j}Gu_i)\,d\Gamma + \int_\Omega (\hat{u}_{i,j}G)_{,j}u_i\,d\Omega$$

$$- \int_\Gamma n_i(\hat{u}_{i,j}Gu_j)\,d\Gamma + \int_\Omega (\hat{u}_{i,j}G)_{,i}u_j\,d\Omega$$

$$- \int_\Gamma n_k(\hat{u}_{i,i}\lambda u_k)\,d\Gamma + \int_\Omega (\hat{u}_{i,i}\lambda)_{,k}u_k\,d\Omega - \int_\Omega \hat{u}_i m\ddot{u}_i\,d\Omega = 0 \qquad (11)$$

Reordering the terms and collecting under common integral signs:

$$\int_\Gamma n_j(\hat{u}_i\sigma_{ij}) - n_j(\hat{u}_{i,j}Gu_i) - n_i(\hat{u}_{i,j}Gu_j) - n_k(\hat{u}_{i,i}\lambda u_k)\,d\Gamma$$

$$+ \int_\Omega (\hat{u}_{i,j}G)_{,j}u_i + (\hat{u}_{i,j}G)_{,i}u_j + (\hat{u}_{i,i}\lambda)_{,k}u_k\,d\Omega - \int_\Omega \hat{u}_i m\ddot{u}_i\,d\Omega = 0 \qquad (12)$$

The indices can now be rewritten so that the unknowns, u_i, may be factored out.

$$\int_\Gamma n_j(\hat{u}_i\sigma_{ij}) - (n_j\hat{u}_{i,j}G + n_j\hat{u}_{j,i}G + n_i\hat{u}_{j,j}\lambda)u_i\,d\Gamma$$

$$+ \int_\Omega [(\hat{u}_{i,j}G)_{,j} + (\hat{u}_{j,i}G)_{,j} + (\hat{u}_{j,j}\lambda)_{,i}]u_i\,d\Omega - \int_\Omega \hat{u}_i m\ddot{u}_i\,d\Omega = 0 \qquad (13)$$

$n_j\sigma_{ij}$ in the first term of Eq. (13) is simply the surface traction and can be defined as $t_i = n_j\sigma_{ij}$. For further simplification, the second

term under the surface integral will be defined as $t^*_i = n_j \hat{u}_{i,j} G + n_j \hat{u}_{j,i} G + n_i \hat{u}_{j,j} \lambda$. So,

$$\int_\Gamma (\hat{u}_i t_i - t^*_i u_i) d\Gamma + \int_\Omega [(\hat{u}_{i,j} G)_{,j} + (\hat{u}_{j,i} G)_{,j} + (\hat{u}_{j,j} \lambda)_{,i}] u_i \, d\Omega - \int_\Omega \hat{u}_i m \ddot{u}_i \, d\Omega = 0 \qquad (14)$$

The first domain integral of Eq. (14) is solved by setting the assumed function terms, commonly called the adjoint opertator, equal to a vector of Dirac Delta functions.

$$(\hat{u}_{i,j} G)_{,j} + (\hat{u}_{j,i} G)_{,j} + (\hat{u}_{j,j} \lambda)_{,i} = \delta(x, x_o) \hat{e}_i \qquad (15)$$

where $\delta(x, x_o)$ is the Dirac delta function and $\hat{e}_i$ is a unit vector. The choice of the Dirac Delta function now allows the domain integral to be integrated exactly. So, substituting Eq. (15) into the domain integral and integrating gives

$$\int_\Omega u_i \delta(x, x_o) \hat{e}_i \, d\Omega = Cu(x) \qquad (16)$$

where x_o is the integration variable and C is a fraction dependent upon the location of integration. If the sigularity due to the Dirac delta

function at $x = x_o$ is integrated about completely, C is equal to one. For the purpose of determining the boundary displacements, the integration will be done on a smooth boundary surface and C will be equal to 1/2.

Making the appropriate substitution, Eq. (14) becomes

$$Cu(x) + \int_{\Gamma} (\hat{u}_i t_i - t^*_i u_i) d\Gamma - \int_{\Omega} \hat{u}_i m\ddot{u}_i \, d\Omega = 0 \qquad (17)$$

In order to evaluate the boundary integrals, $\hat{u}_i$ must be solved for. This task amounts to solving Eq (15). In order to solve Eq. (15), $\hat{u}_i$ is written as $\hat{u}_j = \hat{u}_{jl}(x,x_o)\hat{e}_l$. This second order tensor has the interpretation that the individual elements of $\hat{u}_{jl}$ are the displacements in the Jth direction at the point x_o due to a unit point force acting in the Lth direction, given by $\hat{e}_l$, applied at point x. With this understanding, Eq. (15) can be written

$$(G\hat{u}_{jl})_{,ii} + [(G+\lambda)\hat{u}_{il}]_{,ij} = \delta(x,x_o)\hat{e}_l\hat{e}_j \qquad (18)$$

In 2-D, Danson has solved Eq. (18) to give (6:211-213)

$$\hat{u}_{jl}(x,x_o) = \frac{1+\nu}{4\pi E(1-\nu)}\{(3-4\nu)\ln\frac{1}{r}\,\delta_{jl} + r_j r_l\} \qquad (19)$$

and therefore

$$t^*_{jl} = - \frac{1}{4\pi(1-\nu)r} [n_k r_k \{(1-2\nu)\delta_{jl} + 2r_j r_l\} - (1-2\nu)(r_j n_l - r_l n_j)] \qquad (20)$$

where r is the magnitude of the vector between the point being solved for (x) and the field point (x_o), ν is Poisson's ratio and the r_k's are the direction cosines of r.

When Eqs. (19) and (20) are substituted into Eq. (17) the surface integral can be evaluated given a suitable displacement function. Except for the mass term, which will be handled latter, Eq (17) represents the boundary integral formulation of the elastic-dynamic equation.

2.2 Boundary Element Formulation

Eq. (17) can be discretized by creating boundary elements over the structure. Each integral can be written as a summation of integrals over each element:

$$Cu = \sum_{k=1}^{l} [- \int_{\Gamma_k} \hat{u}_i t_i \, d\Gamma + \int_{\Gamma_k} t^*_i u_i \, d\Gamma] + \int_{\Omega} \hat{u}_i m\ddot{u}_i \, d\Omega \qquad (21)$$

For a 2-D structure each element would appear as a line. Figure 1 shows a simple 2-D beam section paved with 16 boundary elements. By assuming a shape function for each element, the displacement at any point on the element can be written as a function of the nodal point values. For

instance, for a single element as given in Figure 2.

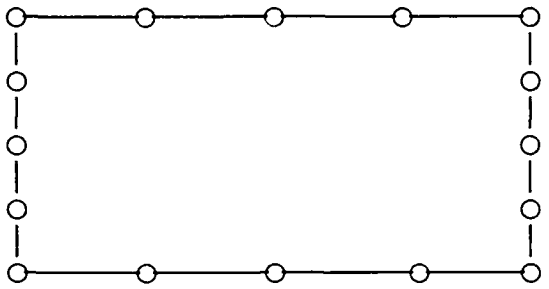


Fig. 1. Beam Section with 16 Elements

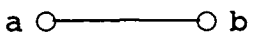


Fig. 2. Single Element

the displacement at any point x on the element can be written as

$$u(x) = u_a \frac{L - x}{L} + u_b \frac{x}{L} \qquad (22)$$

where L is the length of the element and u_a and u_b are the displacements at the a and b nodes. Eq. (22) gives a linear relationship along the element. Higher order elements can be used (i.e. quadratic, etc) depending upon the accuracy requirements. More generally, the functional formulation over each element may be written as

$$u(x) = [\ N_a\ N_b\ ]\{\ u_a\ u_b\ \} \tag{23}$$

or

$$u(x) = [N]\{u\} \tag{24}$$

where $\{u\}$ is a column vector of nodal displacements, $[N]$ is a row of shape functions, and $u(x)$ is the value of the displacement at point x on the element. Surface tractions, t, can be approximated in a similar manner.

With the introduction of elements and shape functions, the boundary integral equation can now be written as a summation of integrals over the individual elements with displacements and tractions as functions of the nodal value vectors.

$$Cu = \sum_{k=1}^{l} [- \int_{\Gamma_k} [N]_k (u_i)_k t^*_i \, d\Gamma + \int_{\Gamma_k} [N]_k (t_i)_k \hat{u}_i \, d\Gamma\ ] - \int_{\Omega} m\ddot{u}_i \hat{u}_i \, d\Omega \tag{25}$$

Since the nodal values under the integral signs are constants they can be taken outside the integrals. In addition, if Eq. (23) is written for each nodal point, the resulting equations can be recast in matrix form.

$$C\{u\} = [G]\{t\} - [S]\{u\} - \{m^*\} \tag{26}$$

where [S] and [G] are matrices of the elemental integrals associated with each nodal point and $\{m^*\}$ are terms associated with the mass integral (yet to be determined). The elemental integrals can be evaluated numerically using Guassian quadrature. The entire discretization process is very similar to that used in finite element methods and is explained in depth by Gipson (7:115-120). Eq. (26) can be further simplified by combining [S] and C:

$$[H]\{u\} = [G]\{t\} - \{m^*\} \tag{27}$$

where $[H] = [S] + C[I]$

2.3 Treatment of the mass

Up until now the mass integral has been ignored, but this integration must be completed in order to solve the equation for dynamic motion. Ahmad and Banerjee have developed a method for handling the mass term using an approximated density function and particular integrals (8:682-694). After applying the method of particular integrals, Eq. (27) becomes

$$[G]\{t\} - [H]\{u\} = w^2([G][Q] - [H][D])[K]\{u\} \tag{28}$$

where w is a frequency term resulting from the assumption of sinusoidal motion and the [Q], [D], and [K] matrices are developed from the particular integral method. The [G] and [H] matrices remain the same as

developed in section 2.2. A detailed explanation of the use of particular integrals with the BEM is contained in Appendix A.

III. Wave Propagation

3.1 Substructure Analyses

At this point Eq. (28) could be applied to an entire structure. If the structure is repetitive, such as a truss structure, it is possible to analyze only one bay of the truss and still calculate the dynamical behavior of the structure.

Figure 3 shows one bay of a two-dimensional truss structure with two attach points on each side. The tractions and displacements associated with each connecting point are also shown.

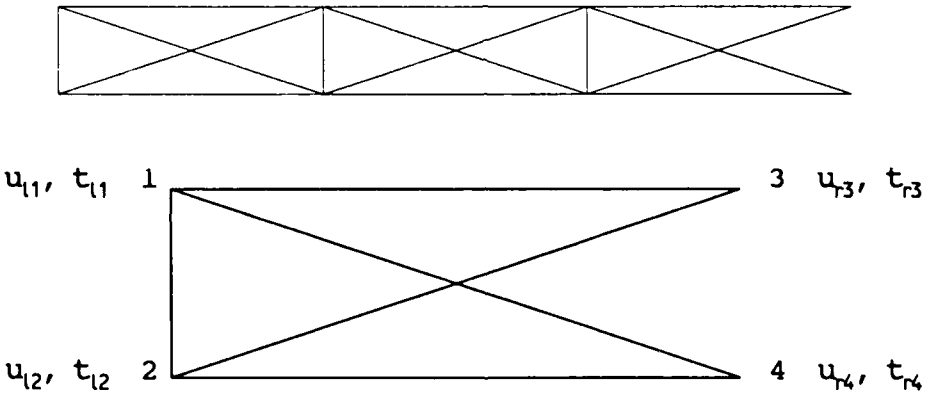


Figure 3 Single Bay of a Repetitive Truss

An equation relating the left and right forces and displacements can be written in terms of a transfer matrix [T]:

$$\{v_r\} = [T]\{v_l\} \tag{29}$$

where

$$\{v_l\} = \{u_{l3}, u_{l4}\ t_{l3}\ t_{l4}\} \tag{30}$$

$$\{v_r\} = \{u_{r1}, u_{r2}, t_{r1}, t_{r2}\} \tag{31}$$

The vector $\{v\}$ at any junction on the truss is called the state vector at that junction. Since each bay has the identical structural characteristics, the transfer matrix for each bay will also be identical. This makes it possible to describe the state at any bay junction using only the substructure transfer matrix. For instance, the forces and displacements at the n^{th} bay can be written as:

$$\{v_r\}_n = [T]^n\{v_l\}_1 \tag{32}$$

Thus, the state vector at any bay junction can be propagated along the structure by use of the transfer matrix.

3.2 Transfer Matrix Derivation

The transfer matrix contains information about the mass and stiffness of the substructure. Using the BEM described in chapter 2, the transfer matrix for the substructure can be derived. Starting with Eq. (28):

$$[G]\{t\}-[H]\{u\} = w^2([G][Q]-[H][D])[K])\{u\} \tag{28}$$

The right hand side of the Eq. (28) can be simplified into a single matrix [m], by carrying out the appropriate multiplication and subtraction:

$$[m] = w^2([G][Q]-[H][D])[K] \tag{33}$$

Substituting Eq. (33) into Eq. (28) gives

$$[G]\{t\}-[H]\{u\} = [m]\{u\} \tag{34}$$

The [H] matrix can now be added to [m] to define a new matrix [M].

$$[M] = [m] + [H] \tag{35}$$

Substituting Eq. (35) into Eq. (34) gives

$$[G]\{t\} = [M]\{u\} \tag{36}$$

Finally, multiplying through by $[M]^{-1}$:

$$[R]\{t\} = \{u\} \tag{37}$$

where

$$[R] = [M]^{-1}[G] \tag{38}$$

[R] can be thought of as the dynamical admittance matrix equation for the boundary nodes of the substructure.

Eq. (37) can be partitioned to separate the left connecting, right connecting, and outer surface nodes as follows.

$$\begin{bmatrix} R_{ll} & R_{lr} & R_{lo} \\ R_{rl} & R_{rr} & R_{ro} \\ R_{ol} & R_{or} & R_{oo} \end{bmatrix} \begin{bmatrix} t_l \\ t_r \\ t_o \end{bmatrix} = \begin{bmatrix} u_l \\ u_r \\ u_o \end{bmatrix} \qquad (39)$$

Where t_o and u_o represent the nodal values that do not lie on a junction. Because there is no contact at the non-junction nodes, the traction on those nodes is identically equal to zero. Therefore, all outer surface terms can be eliminated from Eq. (39) to give

$$\begin{bmatrix} R_{ll} & R_{lr} \\ R_{rl} & R_{rr} \end{bmatrix} \begin{bmatrix} t_l \\ t_r \end{bmatrix} = \begin{bmatrix} u_l \\ u_r \end{bmatrix} \qquad (40)$$

For simplification, Eq. (40) can be written as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} t_l \\ t_r \end{bmatrix} = \begin{bmatrix} u_l \\ u_r \end{bmatrix} \tag{41}$$

At this point, some care must be taken to assure compatibility between the sign convention of the BEM formulation and the transfer matrix equation. Figure 4. shows the sign conventions used for each formulation.

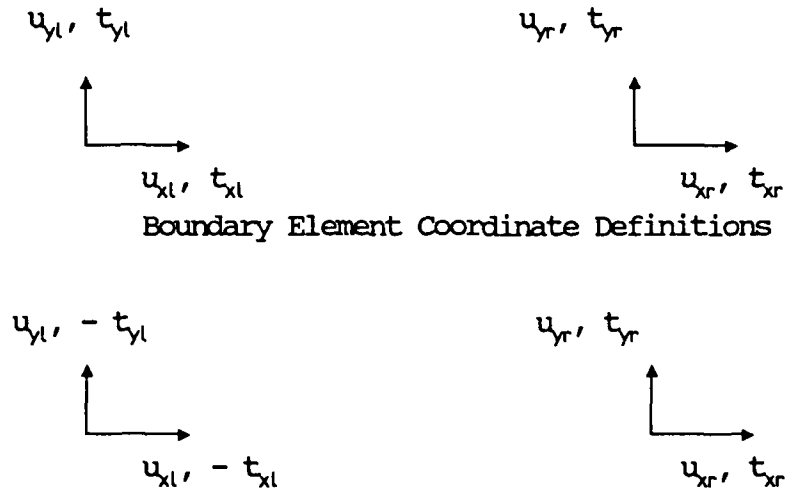


Fig. 4. Coordinate System Definitions

As shown, the sign convention of t_l for the transfer matrix is opposite to that of BEM. Therefore, the left side tractions, in Eq. (41) must be multiplied by -1 in order to satisfy the transfer matrix sign convention. After multiplying t_l by -1 and rearranging, Eq. (41) can be rewritten as

$$\begin{bmatrix} u_r \\ \cdots \\ t_r \end{bmatrix} = \left[\begin{array}{c:c} DB^{-1} - C & DB^{-1}A \\ \hdashline B^{-1} & B^{-1}C \end{array}\right] \begin{bmatrix} u_l \\ \cdots \\ t_l \end{bmatrix} \tag{42}$$

or

$$\{V_r\} = [T]\{V_l\} \tag{43}$$

3.3 Eigenvalues of the Transfer Matrix

Using the approach of Signorelli (4:23-25), wave propagation in a repetitive structure can be represented as:

$$\{V_r\} = \epsilon\{V_l\} \tag{44}$$

Eq (44) shows that the state vector at the right side of the substructure is the same as the state vector at the left multiplied by a factor ϵ. ϵ will generally be complex due to the phase difference

between the response at each end of the substructure. If Eq (44) is combined with Eq (43) the following eigenvalue problem in ϵ is formed.

$$([T] - [I]\epsilon)\{v_l\} = 0 \tag{45}$$

For any given frequency in [T], Eq. (45) will produce a set of eigenvalues. In addition to being complex, ϵ will occur in ϵ and $1/\epsilon$ pairs, corresponding to right and left going waves. For right going waves, the magnitude of ϵ will be less than 1. For left going waves, the magnitude of ϵ will be greater then 1. Eigenvalue magnitudes equal to unity represent a wave mode that will propagate undiminished across the structure. These undiminished wave modes are said to be in a pass band. Eigenvalues with a magnitude other than 1 represent non-propagating wave modes and are said to be in a stop band. Eigenvalue behavior can be represented on the ϵ plane, as shown in Figure 5.

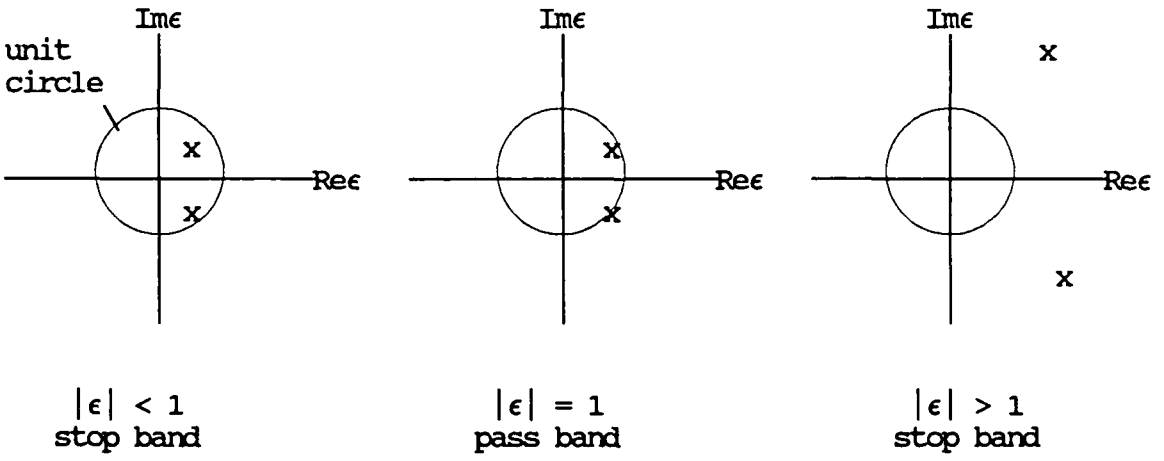


Figure 5. Eigenvalues on the ϵ Plane

The eigenvalues of the transfer matrix can also be represented in exponential form:

$$\epsilon = e^{\alpha L} \tag{46}$$

Where L is the length of the bay. α is complex and can be separated into its real and complex parts:

$$\alpha = a + bi \tag{47}$$

so

$$\epsilon = e^{aL} + e^{biL} \tag{48}$$

or

$$\epsilon = e^{aL} + e^{i(kL + 2\pi)} \tag{49}$$

k is a nondimensional wave number and is related to wavelength y by $k = 2\pi/y$. e^{aL} is an attenuation coefficient and describes the rate of decay as the wave passes through a bay. Negative values of aL indicate a left or negative going wave. The imaginary part of Eq. (49) describes the phase relationship between the state vectors at the left and right side of the bay.

IV. Dynamic Structural Analyses

4.1 Wave Propagation Of a Two-Dimensional Beam

A two-dimensional beam was selected as a baseline structure for demonstrating the implementation of boundary element analysis with wave propagation theory as discussed in chapters 1, 2, and 3. The selection of a simple beam was based on three criteria. First, results are easily compared to well understood continuum models. Second, a beam model is simple to implement yet still provides verification of the analysis procedures. Third, a beam can be thought of as a repetitive structure if divided into several short beam elements.

Figure 6 shows a long beam divided into several sections or "bays".

Figure 6 Sectioned Beam

The properties of the beam were selected as follows:

$E = 1.0 \times 10^6$ psi

$h = 5.0$ in

$m = .10$ lb/in^3

$t = 1.0$ in

$\nu = .2$

where E is the modulus of elasticity, h is height, m is mass density t is thickness and ν is poisson's ratio.

A 10 inch length of beam was selected to represent one bay. The bay was modeled with 30 linear boundary elements as shown in Figure 7.

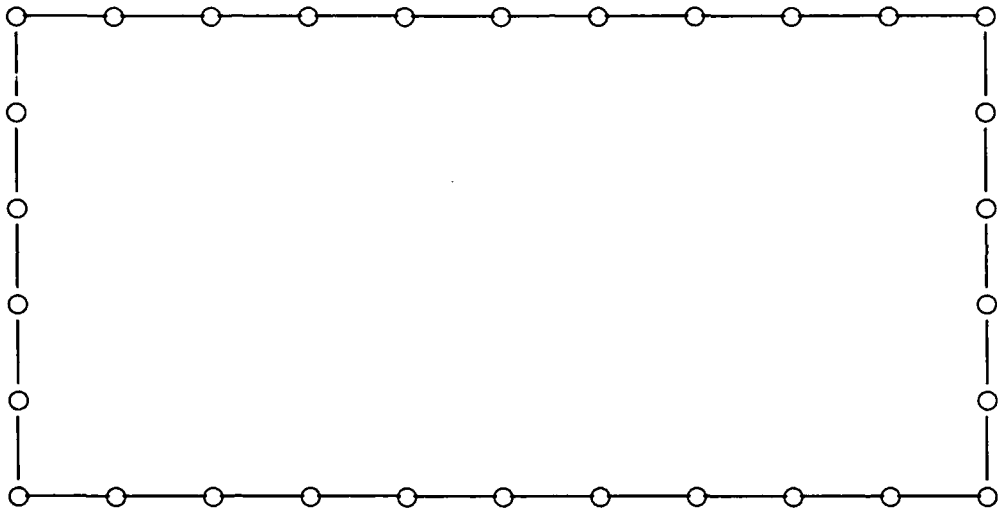


Figure 7 Boundary Elements on Beam Section

A FORTRAN program written by Brebbia for static 2-D structural analyses was modified to calculate the [H] and [G] matrices of Eq.(27) (9:429-438). Additional FORTRAN code was written to implement the particular integral method discussed in section 2.3 (Eq. (28)). The transfer matrix was then calculated using the procedure outlined in chapter 3. The 6 node interface of the beam section produced a 24x24 transfer matrix (12 displacements and 12 tractions). Thus, 24 eigenvalues could be extracted at any given frequency.

4.1.1 Eigenvalue Analysis

The eigenvalues of the transfer matrix were extracted for frequencies from 10 to 200Hz using an EISPACK Fortran solver (10:26-27). At each frequency, four eigenvalues of magnitude 1.0 always appeared. As discussed in section 3.3, eigenvalues of magnitude 1.0 indicate a propagating wave mode. Since complex eigenvalues always appear in complex conjugate pairs, the four propagating eigenvalues represented two wave modes.

Inspection of the eigenvectors revealed that the two propagating wave modes were a bending mode, and a compression-extension mode, as expected for a beam. Figure 8 shows a dispersion curve of the phase of the eigenvalue versus frequency for the bending mode. A similar plot for the compression-extension mode is shown in Figure 9. As shown, the phase angle increases with frequency. This is due to the decreasing wavelengths.

The remaining right going wave modes exhibited stop band behavior ($|\epsilon| < 1$) at all frequencies. A sampling of the stop band eigenvalues is presented in Table 1. The stop band eigenvalues represent localized modes that quickly die out as they travel down the beam. Eigenvalues with a magnitude much smaller than 1 represent localized behavior that does not propagate.

Classical beam theory provides a basis for comparing the results shown in Figures 8 and 9. For bending, simple 2-D beam theory predicts the following behavior for a free-free beam (11:163-166).

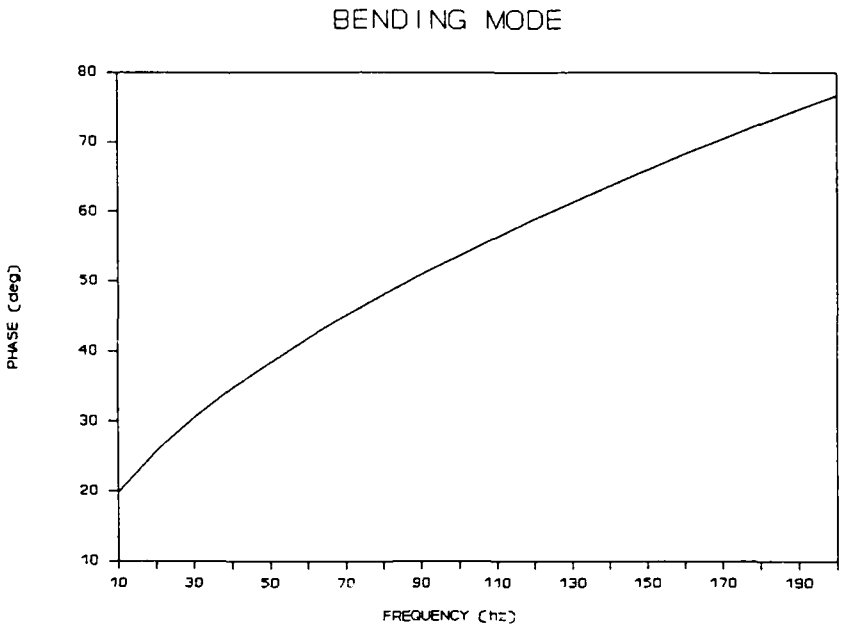


Figure 8 Bending Mode Dispersion Curve

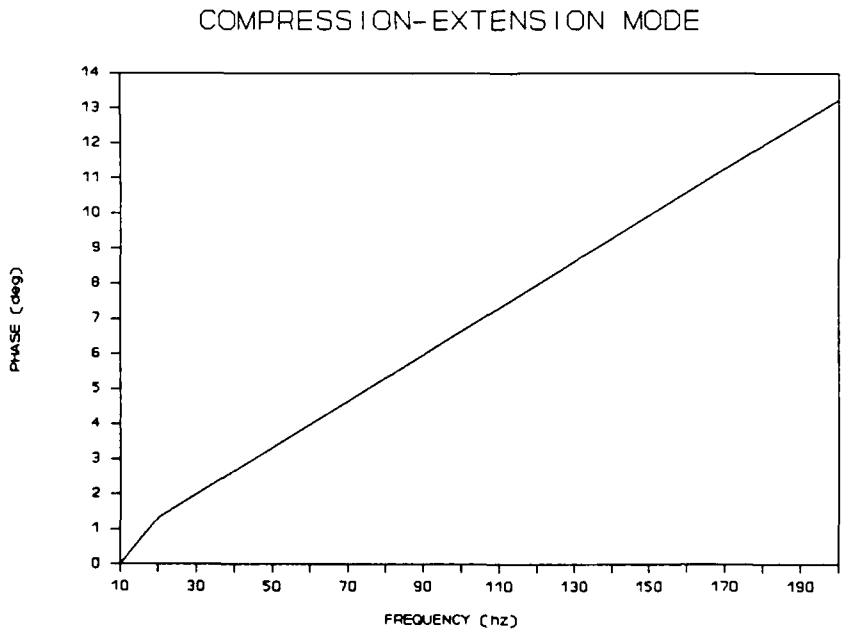


Figure 9 Compression-Extension Mode Dispersion Curve

Table 1 Stop Band Eigenvalues for Beam Bay at 100hz

$\lvert\epsilon\rvert$	Real(ϵ)	Imag(ϵ)	Phase(deg)
.4282E+0	.4282E+0	.0	.0
.9409E-3	-.8778E-3	.3387E-3	-.2110E+2
.1026E-3	-.1026E-3	.0	.0
-.6920E-4	-.2352E-4	.6508E-4	-.7013E+2
.8253E-6	.8253E-6	.0	.0
.3416E-6	-.3416E-6	.0	.0

$$w = (2.5\pi)^2 (EI/mAL^4)^{1/2} \tag{50}$$

where w is the frequency for one wave of a bending mode, I is the moment of inertia, A is the cross sectional area and L is the length. The phase of the wave, β_b, at any point on the beam can be described by a simple ratio:

$$\beta_b = 2\pi x/L \tag{51}$$

where x is the distance of the point from the end of the beam and 2π is simply the phase of one complete wave. Solving Eq. (50) for 1/L

$$1/L = \{(w/(2.5\pi)^2 (mA/EI)^{1/2}\}^{1/2} \tag{52}$$

and multiplying through by $2\pi x$

$$2\pi x/L = 2\pi x\{(w/(2.5\pi)^2 (mA/EI)^{1/2}\}^{1/2} \tag{53}$$

gives the theoretical phase angle:

$$\beta_b = 2\pi x\{(w/(2.5\pi)^2(m\bar{A}/EI)^{1/2})^{1/2} \quad (54)$$

A similar development for the theoretical compression-extension phase angle gives.

$$\beta_{ce} = (xw/\pi)(m/E)^{1/2} \quad (55)$$

If the length of the beam bay is used for x in Eq. (54) and (55), a comparison to calculated results can be made. This comparison is shown in Figures 10 and 11. As shown in Figure 10, there is closer agreement with bending theory at lower frequencies. As the frequency increases, the BEM is less able to model the bending behavior. This disparity at higher frequencies is not uncommon for discreet formulations and is similar to results obtained by Signorelli using FEM (4:30). Some of the disparity may be due to the use of linear elements in the formulation. Figure 11 shows nearly perfect agreement with theory for the compression extension mode. This should be expected, because compression-extension deformations are more easily modeled with linear elements.

4.1.2 Eigenvector Mode Shapes

For each eigenvalue, a corresponding eigenvector {V} can be calculated and used to generate a plot of the deformed structure. Each

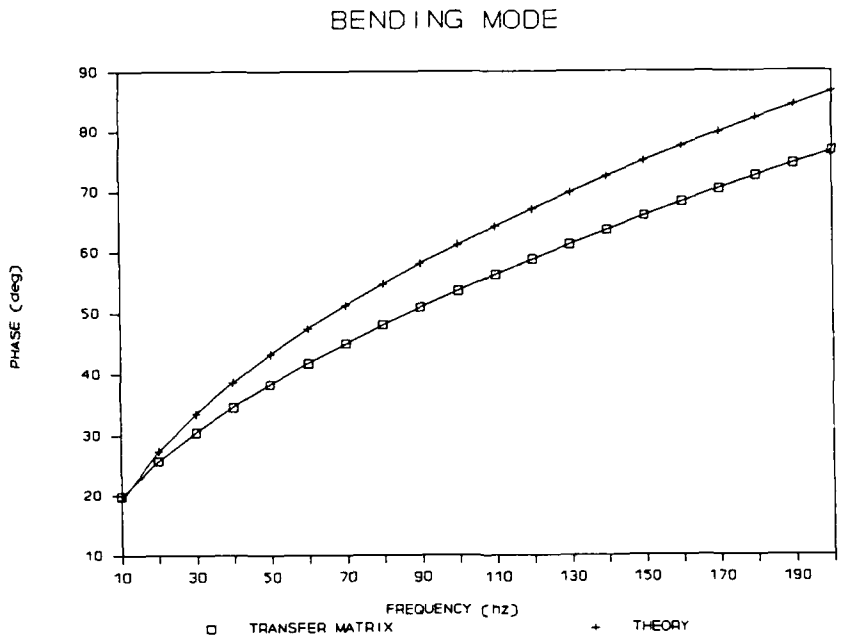


Figure 10 Theoretical vs Analytical Bending Mode Dispersion Curve

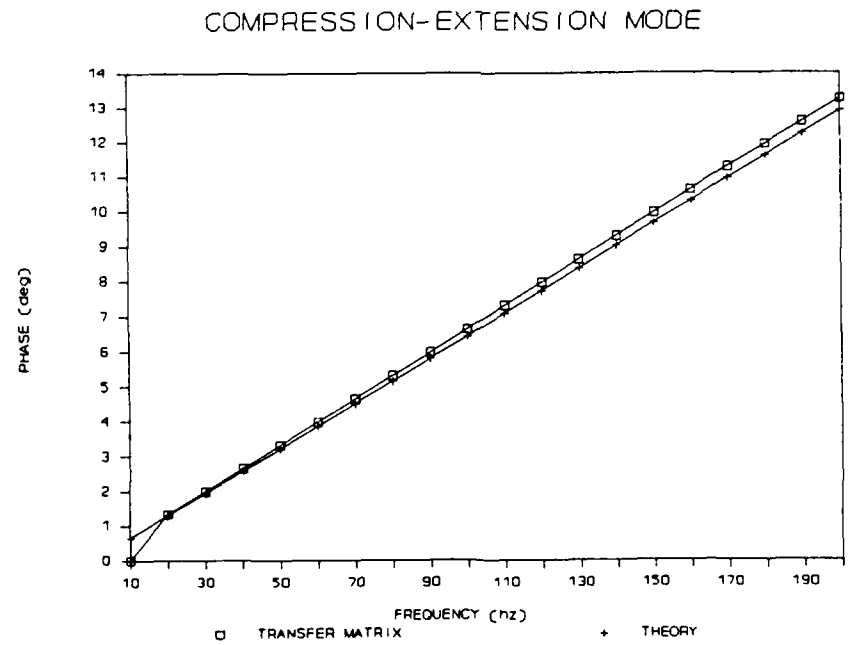


Figure 11 Theoretical vs Analytical Compression-Extension Mode Dispersion Curve

eigenvector contains values of the displacements and forces on the left hand side of the beam section:

$$\{v_r\} = \{u_r, t_r\} \tag{56}$$

In turn, Eq. (44) can be used to calculate the state vector on the right hand side of the section.

$$\{v_r\} = \epsilon\{v_l\} \tag{57}$$

The right hand side of one section now becomes the left hand side of the adjacent section, allowing the state vector to be propagated along the beam.

$$\{v_r\}_n = \epsilon\{v_r\}_{n-1} \tag{58}$$

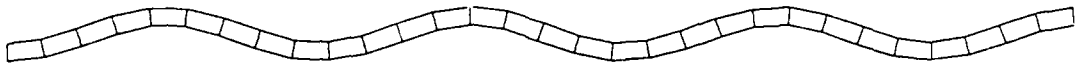
were n is a bay number. Recovery of the deflections on the outer surface of any bay is accomplished by expanding Eq. (39) to get

$$u_{on} = [R_{ol}]\{t_l\}_n + [R_{or}]\{t_r\}_n \tag{59}$$

By adding the real part of the deflections to the original node locations, the mode shapes can plotted.

Figure 12 shows plots of the bending mode for several frequencies. Deflections are multiplied by an appropriate scaling factor to accentuate the mode shape. In addition, the plots of Figure 12 were generated using only the deflections of the 6 junction nodes propagated to the right and connected by straight lines at the outer beam surface. A closer look at the bending behavior for a single bay, with outer surface node deflections included, is shown in Figure 13.

40hz Bending Mode



60hz Bending Mode

80hz Bending Mode

100hz Bending Mode

Figure 12 Bending Modes at Selected Frequencies

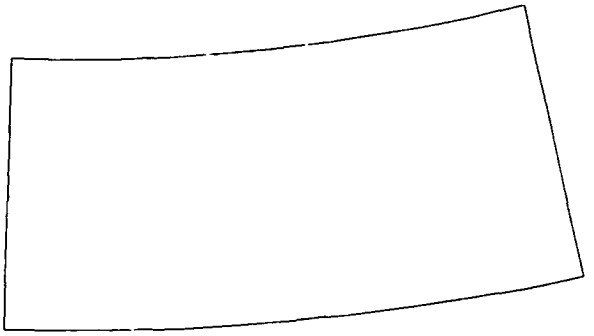


Figure 13 Bending Mode For Single Bay at 60hz

The number of bays necessary to complete one cycle of a wave depends upon the phase angle of the eigenvalue. For instance, at 80hz the phase angle is 48.1 degrees - requiring approximately 7.5 bays to complete one 360 degree cycle.

Plots of the compression-extension mode, shown in Figure 14, are not as dramatic as bending, due to the small phase angles and relatively small deflections. As shown, for 200 hz the compression-extension mode would require approximately 27 bays to complete a cycle.

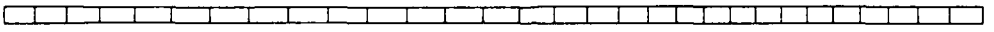


Figure 14 Compression-Extension Mode at 200hz

Figure 15 demonstrates the behavior of a stop band mode at 60hz with ϵ = .519 + 0.i. As predicted, the wave is quickly attenuated. An additional non attenuating wave, with ϵ = -.92E-3 + .21E-3i is shown in Figure 16 for one bay. This localized wave does not even have a

noticeable effect on the right side of the bay due to the small value of $|\epsilon|$.

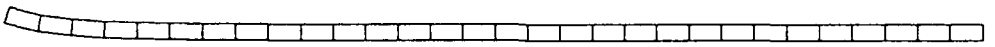


Figure 15 Stop Band Mode at 60hz

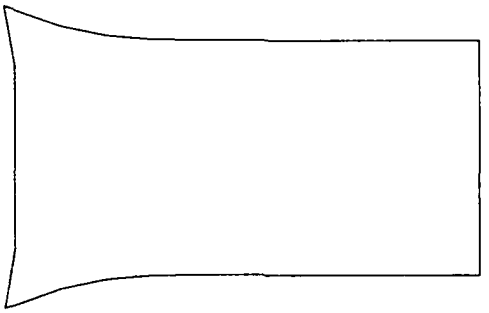


Figure 16 Localized Deformation at 50hz

It should be noted that any one wave mode does not represent the total behavior of a structure. The dynamics of any particular structure would be a time dependent linear combination of all eigenvectors.

4.2 Natural Frequencies

Two methods have been proposed for using the transfer matrix to identify natural frequencies. The first method, proposed by Signerorelli requires calculation of a global transfer matrix $[T]_g$ (4:22). For a structure with n bays, Eq. (43) can be applied

sequentially to each bay from left to right to obtain

$$\{v_r\}_n = [T]^n\{v_l\}_1 \tag{60}$$

so

$$[T]_g = [T]^n \tag{61}$$

and

$$\{v_r\}_n = [T]_g\{v_l\}_1 \tag{62}$$

Now, boundary conditions can be applied. For a cantilever beam, the boundary conditions would be

$$\{v_r\}_n = \{u_r, 0\}_n \tag{63}$$

for no tractions on the free end. The secured end would have zero displacements:

$$\{v_l\}_1 = \{0, t_l\}_1 \tag{64}$$

So, after droping subscripts on the vectors, Eq. (62) becomes

$$\{u_r, 0\} = [T]_g\{0, t_l\} \tag{65}$$

$[T]_g$ can be partitioned to give

$$\begin{bmatrix} u_r \\ \cdots \\ 0 \end{bmatrix} = \left[\begin{array}{c:c} T_1 & T_2 \\ \hdashline T_4 & T_3 \end{array}\right]_g \begin{bmatrix} 0 \\ \cdots \\ t_l \end{bmatrix} \tag{66}$$

The bottom row of Eq. (66) gives

$$0 = [T_3]\{t_l\} \tag{67}$$

The only non-trivial way for Eq. (67) to hold true is for the determinant of $[T_3]$ to equal zero. Therefore a plot of Det($[T_3]$) versus frequency should reveal the natural frequencies. Unfortunately, the computation of $[T]^n$ for any significant number of bays will generally exceed the computational limit of the computer. An attempt at computing det($[T_3]$) for just three bays of the beam section resulted in numerical overflow.

Von Flotow proposes a second method for determining natural frequencies (3:516). This method uses the eigenvectors of the transfer matrix along with the boundary conditions to calculate a scattering matrix for each end of the structure. By using the scattering matrices

along with the eigenvalues of the transfer matrix another determinant problem is set up by virtue of the phase closure principle. Again, this method proves numerically unstable for any sizeable matrices. The application of this method on the beam section proved unsuccessful.

Although use of BEM with wave propagation theory does not easily lend itself to identifying natural frequencies, there are time domain techniques that can be used with wave theory to model the real-time behavior of structures. Von Flotow has demonstrated the successful use of a time domain technique for the traveling wave analysis of a structural network (3:518).

V. Summary

5.1 Conclusions

The purpose of this effort was to demonstrate the use of boundary element methods in performing a wave propagation analysis on periodic structures. The combination of these two analysis techniques may be useful in describing the dynamic behavior of repetitive structures such as those planned as orbiting space platforms. Past work in the area of wave propagation in structures has been limited to using continuum or finite element structural models.

Although limited in its scope, this effort has produced several important results. First, a computer code was produced that formulates the transfer matrix of a 2-D isotropic element using boundary element techniques. This code can be used to find the spatial eigenvalues and eigenvectors of a periodic structure. Second, results from the analysis of a baseline beam section were in close agreement with theoretical results. The disparity of results for bending at higher frequencies was most likely due to the use of linear elements in the formulation of the transfer matrix. Disagreement of data at higher frequencies is not uncommon in most discreet formulations, including FEM. Third, although BEM matrices are generally smaller then those of equivalent FEM, their dense nature can produce problems in some numerical techniques such as finding determinants. Finally, the demonstration of this method in 2-D holds promise for its use in more complex structural descriptions.

5.2 Recommendations

The following suggestions are made for future work in this area.

(1) Investigate the use of BEM for describing the behavior of joints in periodic structures.

(2) Develop a code for formulating the transfer matrix of a three dimensional structure using boundary element techniques.

(3) Investigate the inclusion of damping terms in the boundary element equation.

(4) Combine BEM and FEM, where appropriate, to model large truss like structures. For instance, FEM might best be suited to rod and beam elements while BEM could be used for joints and other solid body members of the structure.

(5) Compare the computational efficiency between BEM and FEM in formulating the transfer matrix of different structures.

(6) Use BEM in describing nonlinear behavior such as plastic deformation.

(7) Develop an equivalent transfer matrix description of nonsymmetric 2-D structures such as triangles.

Appendix A: Ahmad and Banerjee's Particular Integral Method

The particular integral method uses the fact that the solution to the elastic dynamic equations can be written as the sum of a complementary function, u^c_j, and a particular integral, u^p_j. The complementary solution satisfies

$$Gu^c_{j,ii} + (G+\lambda)u^c_{i,ji} = 0 \tag{68}$$

while the particular integral satisfies

$$Gu^p_{j,ii} + (G+\lambda)u^p_{i,ji} = \rho\omega^2 u_j \tag{69}$$

Eq. (69) still contains the unknown displacement field u_j. But, u_j can be described using an unknown fictitious density function, f, and a known function, C, in a manner similar to using shape functions:

$$u_j = \sum_{n=1}^{l} C^n_{jk} f^n_k \tag{70}$$

A simple function that can be selected for C is

$$C_{ik} = \delta_{ik}(R-r) \tag{71}$$

where R is the largest distance between two points on the body and r is the magnitude of the vector from the field point to the source point. Eq. (69) can now be written as

$$G u^p_{j,ii} + (G + \lambda) u^p_{i,ji} = \rho\omega^2 \sum_{n=1}^{l} C^n_{jk} f^n_k \tag{72}$$

u^p_j can now be chosen as any function that satisfies Eq. (72) and can be represented as

$$u^p_j = \sum_{n=1}^{l} D^n_{jk} f_k \tag{73}$$

were D_{jk} is found to be

$$D_{jk} = \frac{\rho\omega^2}{G} \left(\left(\frac{(9-10\nu)r}{90(1-\nu)} - \frac{(1-2\nu)R}{6-8\nu} \right) \delta_{jk} r^2 - \frac{1}{30(1-\nu)} r_j r_k r \right) \tag{74}$$

The surface traction, t^p_j, related to the displacement field of Eq. (74) is defined as

$$t^p_j = \sum_{n=1}^{l} Q^n_{jk} f^n_k \tag{75}$$

Q can be found using Eq. (74) along with the strain-displacement relationships, and is given by

$$Q_{jk} = \pi w^2 \left(\left(\frac{(5\nu-1)r}{15(1-\nu)} - \frac{2\nu R}{3-4\nu} \right) r_k n_j + \left(\frac{(4-5\nu)r}{15(1-\nu)} - \frac{(1-2\nu)R}{3-4\nu} \right) r_j n_k \right.$$

$$\left. + \left(\left(\frac{(4-5\nu)r}{15(1-\nu)} - \frac{(1-2\nu)R}{3-4\nu} \right) \delta_{jk} - \frac{1}{15(1-\nu)} \right) r_i n_i \right\} \qquad (76)$$

If Eq. (27) is now assumed to satisfy the complementary solution

$$[G]\{t^c\} - [H]\{u^c\} = 0 \qquad (77)$$

then

$$[G]\{t\} - [H]\{u\} = [G]\{t^p\} - [H]\{u^p\} \qquad (78)$$

Substituting Eq. (73) and (75) into (78), gives

$$[G]\{t\} - [H]\{u\} = w^2([G][Q] - [H][D])\{f\} \qquad (79)$$

$\{f\}$ can be found from Eq. (70)

$$\{f\}_i = \delta_{ij}[K]\{u\}_j \qquad (80)$$

Where [K] is the inverse of the symmetric matrix formed by the function (R-r) applied at each nodal point. Altogether, Eq. (79) becomes

$$[G]\{t\} - [H]\{u\} = w^2([G][Q] - [H][D])[K]\{u\} \qquad (81)$$

Appendix B: Program for solution of Two Dimensional Transfer Matrices by the Boundary Element Method

This program reads an input file that describes the material properties and geometry of the problem. Below is a listing of the input file used for the beam section of chapter 4 along with an explanation of the fields.

Input File Format

First line: Title of Problem

Second line: 0,(number of nodes),(number of elements),(number of junction nodes),0,(type of problem;1=plane strain 2=plane stress),0

Third line: (E*thickness),(poisson's ratio),(p*thickness),(height),(length),(I)

Node position lines: (node number),(X coordinate),(Y coordinate),(blank),0

Note: The junction nodes must be numbered consecutively starting with node 1 on the right side and continuing on to the left side. Exterior surfaces are numbered last.

Element connectivity lines: (element number),(node 1),(node 2)

Note: The direction from node 1 to node 2 must be counterclockwise on exterior surfaces and clockwise on interior surfaces.

Last line: 0,0

Sample Input File

```
RECTANGULAR BOX WITH 5X10X1 DIMENSION
0,30,30,6,0,2,0
1.D+6,.2,3.2D-4,5,10,10.041667
1,0.,5.,,0
2,0.,4.,,0
3,0.,3.,,0
4,0.,2.,,0
5,0.,1.,,0
6,0.,0.,,0
7,10.,5.,,0
8,10.,4.,,0
9,10.,3.,,0
10,10.,2.,,0
11,10.,1.,,0
12,10.,0.,,0
13,1.,5.,,0
14,2.,5.,,0
15,3.,5.,,0
16,4.,5.,,0
17,5.,5.,,0
18,6.,5.,,0
19,7.,5.,,0
20,8.,5.,,0
21,9.,5.,,0
22,1.,0.,,0
23,2.,0.,,0
24,3.,0.,,0
25,4.,0.,,0
26,5.,0.,,0
27,6.,0.,,0
28,7.,0.,,0
29,8.,0.,,0
30,9.,0.,,0
1,13,1
2,14,13
3,15,14
4,16,15
5,17,16
6,18,17
7,19,18
8,20,19
9,21,20
10,7,21
11,6,22
12,22,23
13,23,24
14,24,25
```

```
15,25,26
16,26,27
17,27,28
18,28,29
19,29,30
20,30,12
21,1,2
22,2,3
23,3,4
24,4,5
25,5,6
26,8,7
27,9,8
28,10,9
29,11,10
30,12,11
0,0
```

Program Listing

```
C     PROGRAM FOR CALCULATION OF TWO DIMENSIONAL
C     ELASTODYNAMIC TRANSFER MATRICES BY THE
C     BOUNDARY ELEMENT METHOD
      implicit real*8(A-H,O-Z)
      COMMON /RW/ IRE,IWR
      COMMON /A/ D(2,2),XI(6,3),W(6,3),IDUP(50),INC(50,2),C(50),
     *S(50,3),ISYM(100),X(100),Y(100),IFIP(100),AK(100,100),P(100),
     *XM(100),AG(100,100)
     *,PM(100,100),DM(100,100),TM(100,100)
      INTEGER DMN
      REAL*8 DM1(100,100),TM1(100,100),KM(100,100),WKAREA(2600)
     *,LENTH,INERT
C     INPUT
      OPEN(8,FILE='INPUT.IN',STATUS='OLD')
      OPEN(9,FILE='OUTPUT.OUT',STATUS='NEW')
      IRE=8
      IWR=9
      DMN=100
      IDGT=0
      CALL INPUT(NE,NN,NP,IPL,PO,NN2,NT,NTL,C1,C2,
     *C3,C4,C5,C6,C7,C8,C9,C10,C11,IDSYM,XSYM,YSYM,INFB
     *,RO,E,GM,RMAX,area,lenth,inert)
C     COMPUTE MATRICES H,G,D,F AND K
      CALL MATRX(NE,NN,NN2,NT,C1,C2,C3,C4,C5,C6,C7,C8,
     *C9,C10,C11,PO,IDSYM,XSYM,YSYM,INFB,IFA,NIF,RO,GM,RMAX)
      CALL MATMULT(NN2,AK,DM,DM1)
      CALL MATMULT(NN2,AG,TM,TM1)
      CALL MATSUB(NN2,TM1,DM1,DM)
```

```
      CALL INVERT(PM,TM,NN,DMN)
      CALL MATEXP(NN,TM,KM)
      CALL MATMULT(NN2,DM,KM,PM)
      CALL FRESWP(NN2,NTL,PM,AK,AG,RO,E,area,lenth,inert
     *,INC,X,Y)

    4 STOP
      END
C
C
C
C
C
      SUBROUTINE INPUT(NE,NN,NP,IPL,PO,NN2,NT,NTL,C1,C2,
     *C3,C4,C5,C6,C7,C8,C9,C10,C11,IDSYM,XSYM,YSYM,INFB
     *,RO,E,GM,RMAX,AREA,LENTH,INERT)
      implicit real*8(A-H,O-Z)
      REAL*8 AREA,LENTH,INERT
      COMMON /RW/ IRE,IWR
      COMMON /A/ D(2,2),XI(6,3),W(6,3),IDUP(50),INC(50,2),C(50),
     *S(50,3),ISYM(100),X(100),Y(100),IFIP(100),AK(100,100),P(100),
     *XM(100),AG(100,100)
     *,PM(100,100),DM(100,100),TM(100,100)
      CHARACTER TITLE*70
      WRITE(IWR,1)
    1 FORMAT(//////////,24X,'* * *  B O U N D A R Y  E L E M E N T  M E
     *T H O D  A P P L I E D  T O  * * *',//,24X,'* * * P L A N E  E L
     *A S T O S T A T I C  P R O B L E M S          * * *',///)
C     TITLE OF PROBLEM
      READ(IRE,2)TITLE
    2 FORMAT(A70)
      WRITE(IWR,2)TITLE
C     GENERAL INFORMATION ABOUT THE PROBLEM
      READ(IRE,*)INFB,NE,NN,NTL,NP,IPL,IDSYM
      READ(IRE,*)E,PO,RO,AREA,LENTH,INERT
    3 FORMAT(I1,I4,4I5,2F10.0)
      IF(INFB.EQ.0)GO TO 60
      WRITE(IWR,61)
   61 FORMAT(//,13X,'* INFINITE BOUNDARY *')
   60 WRITE(IWR,4)NE,NN,NP,IPL,IDSYM,E,PO
    4 FORMAT(//,15X,'NO. ELEMENTS =',I5,//,15X,
     *'NO.      NODES =',I5,//,15X,'NO.    POINTS =',I5,//,15X,'PROBL.
     *   TYPE =',I5,//,15X,'SYMME.   TYPE =',I5,///,15X,'MATERIAL
     *PROPERTIES',//,15X,'E     =',F10.0,
     *//,15X,'POISSON =',F15.5,///,30X,'COORDINATES OF BOUNDRY NODES',
     *//,12X,'NODE',14X,'X',15X,'Y',12X,'DOUBLE',/)
      NN2=NN*2
      NT=NN+NP
C     NODES AND POINTS COORDINATES
      DO 5 I=1,NN
```

```
      READ(IRE,*)K,X(K),Y(K),IDUP(K),ISYM(K)
    6 FORMAT(I5,2F10.0,2I5)
      IF(IDUP(K).EQ.0)GO TO 5
      J=IDUP(K)
      IDUP(J)=K
      X(K)=X(J)
      Y(K)=Y(J)
    5 CONTINUE
      DO 63 K=1,NN
      IF(IDUP(K).NE.0)GO TO 62
      WRITE(IWR,7)K,X(K),Y(K)
      GO TO 63
   62 WRITE(IWR,16)K,X(K),Y(K),IDUP(K)
   16 FORMAT(10X,I5,5X,F15.4,1X,F15.4,7x,I5)
   63 CONTINUE
    7 FORMAT(10X,I5,5X,F15.4,1X,F15.4)
      IF (NP.EQ.0)GO TO 9
      WRITE(IWR,8)
    8 FORMAT(//,30X,'COORDINATES OF INTERNAL POINTS',//,11X,'POINT',
     *14X,'X',15X,'Y',/)
      K=NN+1
      READ(IRE,*)(J,X(J),Y(J),ISYM(J),JJ=K,NT)
   14 FORMAT(I5,2F10.0,5X,I5)
      WRITE(IWR,7)(J,X(J),Y(J),J=K,NT)
C     NODES AND POINTS AT SYMMETRY LINES
    9 IF(IDSYM.EQ.0)GO TO 49
      WRITE(IWR,42)
   42 FORMAT(//,30X,'BOUNDRY NODES AND INTERNAL POINTS AT SYMMETRY
     * LINE',//,12X,'L. X',12X,'L. Y',/)
      DO 43 K=1,NT
      IF(ISYM(K).EQ.0)GO TO 43
      IZZ=ISYM(K)
      GO TO (44,45,46),IZZ
   44 YSYM=Y(K)
      WRITE(IWR,47)K
   47 FORMAT(10X,I5)
      GO TO 43
   45 XSYM=X(K)
      WRITE(IWR,48)K
   48 FORMAT(26X,I5)
      GO TO 43
   46 WRITE(IWR,50)K,K
   50 FORMAT(10X,I5,11X,I5)
   43 CONTINUE
C     ELEMENT CONNECTIVITY
   49 WRITE(IWR,10)
   10 FORMAT(//,30X,'ELEMENT CONNECTIVITY',//,13X,'EL',13X,'N. 1',
     *12X,'N. 2',14X,'L',/)
      DO 11 I=1,NE
      READ(IRE,*)K,INC(K,1),INC(K,2)
```

```
   12 FORMAT(3I5)
      II=INC(K,1)
      IF=INC(K,2)
   11 C(K)=SQRT((X(IF)-X(II))**2+(Y(IF)-Y(II))**2)
      WRITE(IWR,13)(I,INC(I,1),INC(I,2),C(I),I=1,NE)
   13 FORMAT(10X,I5,11X,I5,11X,I5,5X,F15.4)
C     CONSTANTS
      GM=E/(2.*(1.+PO))
      C11=PO
      IF(IPL-1)40,40,41
   40 PO=PO/(1.+PO)
      C11=0.
   41 C2=3.-4.*PO
      C3=1./((1.-PO)*12.56637062)
      C4=1.-2.*PO
      C6=2.*C3*GM
      C7=1.-4*PO
      C1=C3/(2.*GM)
      C5=C1/2.
      C8=2.*GM/(1.-PO)
      C9=PO/(1.-PO)
      C10=(2.-PO)/(1.-PO)
C     BOUNDRY VALUES PRESCRIBED
      DO 19 I=1,NN2
      P(I)=0
   19 IFIP(I)=0
      READ(IRE,*)NFIP,NDFIP
   20 FORMAT(2I5)
      WRITE(IWR,21)NFIP,NDFIP
   21 FORMAT(//,15X,'NO. DISPL. PRESC. =',I5,//,15X,'NO. TRACT. PRESC.
     * =',I5,///,15X,'DISPLACEMENTS
     *',//,12X,'NODE',14X,'U',15X,'V',/)
      IF(NFIP.EQ.0.)GO TO 22
      DO 23 I=1,NFIP
      READ(IRE,*)K,P(2*K-1),P(2*K),IFIP(2*K-1),IFIP(2*K)
   24 FORMAT(I5,2F10.0,2I5)
      IND=IFIP(2*K-1)+2*IFIP(2*K)
      GO TO (25,26,27),IND
   25 WRITE(IWR,28)K,P(2*K-1)
   28 FORMAT(10X,I5,5X,F15.4)
      GO TO 23
   26 WRITE(IWR,29)K,P(2*K)
   29 FORMAT(10X,I5,21X,F15.4)
      GO TO 23
   27 WRITE(IWR,30)K,P(2*K-1),P(2*K)
   30 FORMAT(10X,I5,5X,F15.4,1X,F15.4)
   23 CONTINUE
   22 IF(NDFIP.EQ.0)GO TO 31
      WRITE(IWR,34)
   34 FORMAT(//,15X,'TRACTIONS',//,12X,'NODE',13X,'PX'14X,'PY',/)
```

```
      DO 32 I=1,NDFIP
      READ(IRE,*)K,P(2*K-1),P(2*K)
   33 FORMAT(I5,2F10.0)
   32 WRITE(IWR,30)K,P(2*K-1),P(2*K)
C     COMPUTE LARGEST DISTANCE BETWEEN POINTS
   31 RMAX =0.
      DO 90 I=1,NN
      DO 90 J=1,NN
      XX=X(I)-X(J)
      YY=Y(I)-Y(J)
      RMA=SQRT(XX**2+YY**2)
      IF(RMA.GE.RMAX) RMAX=RMA
   90 CONTINUE
C     INTEGRATION POINTS
      XI(1,3)=-0.932469514203152
      XI(2,3)=-0.661209386466265
      XI(3,3)=-0.238619186083197
      XI(4,3)=-XI(3,3)
      XI(5,3)=-XI(2,3)
      XI(6,3)=-XI(1,3)
      W(1,3)=0.171324492379170
      W(2,3)=0.360761573048139
      W(3,3)=0.467913934572691
      W(4,3)=W(3,3)
      W(5,3)=W(2,3)
      W(6,3)=W(1,3)
      XI(1,2)=-0.861136311594053
      XI(2,2)=-0.339981043584856
      XI(3,2)=-XI(2,2)
      XI(4,2)=-XI(1,2)
      W(1,2)=0.347854845137454
      W(2,2)=0.652145154862546
      W(3,2)=W(2,2)
      W(4,2)=W(1,2)
      XI(1,1)=-0.577350269189626
      XI(2,1)=-XI(1,1)
      W(1,1)=1.
      W(2,1)=1.
      RETURN
      END
C
C
C
C
C
C
C
      SUBROUTINE MATRX(NE,NN,NN2,NT,C1,C2,C3,C4,C5,C6,C7,C8,
     *C9,C10,C11,PO,IDSYM,XSYM,YSYM,INFB,IFA,NIF,RO,GM,RMAX)
      implicit real*8(A-H,O-Z)
```

```
      COMMON /A/ D(2,2),XI(6,3),W(6,3),IDUP(50),INC(50,2),C(50),
     *S(50,3),ISYM(100),X(100),Y(100),IFIP(100),AK(100,100),P(100),
     *XM(100),AG(100,100)
     *,PM(100,100),DM(100,100),TM(100,100)
      REAL*8 NORM1(100),NORM2(100),NORM(2)
      COMMON /A4/ H(3,4),G(3,4),HL(3,4),GL(3,4),NORM1,NORM2
     *,NORM,RM(2)
C     KRONECKER DELTA
      D(1,1)=1.
      D(2,2)=1.
      D(1,2)=0.
      D(2,1)=0.
C     CLEAR ARRAYS
      DO 1 I=1,NN2
      XM(I)=0.
      DO 1 J=1,NN2
      AK(I,J)=0.
    1 AG(I,J)=0.
C     COMPUTE PARAMETERS FOR SYMMETRY LOOP
      IFA=1
      NIF=1
      IF(IDSYM.EQ.1)IFA=2
      IF(IDSYM.NE.2)GO TO 60
      IFA=3
      NIF=2
   60 IF(IDSYM.EQ.3)IFA=4
C     TEST FOR INFINITE BOUNDRY
c     IF(INFB.EQ.0)GO TO 90
c     DO 91 I=1,NN2
c     IF(IFIP(I).NE.0)GO TO 92
c     A(I,I)=1
c     GO TO 91
c 92  XM(I)=-P(I)
c 91  CONTINUE
C     SYMMETRY LOOP
   90 DO 2 ISY=1,IFA,NIF
C     COMPUTE CHANGE SIGN CONTROLLING PARAMETERS
      GO TO (70,71,71,73),ISY
   71 IIS=4-ISY
      IFS=IIS
      GO TO 70
   73 IIS=1
      IFS=2
C     LOOP OVER BOUNDRY NODES
   70 DO 2 I=1,NN
      XS=X(I)
      YS=Y(I)
      IF(ISY.EQ.2.OR.ISY.EQ.4)YS=2.*YSYM-YS
      IF(ISY.GE.3)XS=2.*XSYM-XS
C     GENERATE MATRIX H AND G
```

```
C     NOTE:  COLUMNS OF A THAT CORRESPOND TO G ARE MULTIPIED
C     BY C8 TO AVOID ROUNDOFF ERRORS IN THE SOLUTION
      DO 10 J=1,NE
      II=INC(J,1)
      IF=INC(J,2)
      ICOD=1
      IF(ISY.NE.1.AND.ISYM(I).NE.(ISY-1))GO TO 6
      IF(I.EQ.II.OR.I.EQ.IDUP(II))ICOD=2
      IF(I.EQ.IF.OR.I.EQ.IDUP(IF))ICOD=3
    6 CALL FUNC(ICOD,J,C1,C2,C3,C4,C5,C6,C7,PO,II,IF,XS,YS,ISY,IIS,
     *IFS)
      DO 10 K=1,2
      JJ=2*(I-1)+K
      M=0
      DO 10 NX=1,2
      DO 10 NV=1,2
      M=M+1
      IC=2*INC(J,NX)+NV-2
C     IF(IFIP(IC).NE.0)GO TO 67
      AK(JJ,IC)=AK(JJ,IC)+H(K,M)
C     XM(JJ)=XM(JJ)+G(K,M)*P(IC)
C     GO TO 68
   67 AG(JJ,IC)=AG(JJ,IC)+G(K,M)
C     XM(JJ)=XM(JJ)-H(K,M)*P(IC)
C     COMPUTE REMAINING COEFFICIENTS BY APPLYING RIGID BODY TRANSLATIONS
   68 GO TO (61,62,63,64),ISY
   62 IF(NV-2)61,64,61
   63 IF(NV-1)61,64,61
   64 H(K,M)=-H(K,M)
C  61 IF(IFIP(JJ+NV-K).NE.0)GO TO 69
   61 AK(JJ,JJ+NV-K)=AK(JJ,JJ+NV-K)-H(K,M)
      GO TO 10
C  69 XM(JJ)=XM(JJ)+H(K,M)*P(JJ+NV-K)
   10 CONTINUE
    2 CONTINUE
C     COMPUTE MASS MATRIX USING PARTICULAR INTEGRALS
      CM1=(9-10*PO)/(90-90*PO)
      CM2=(1-2*PO)/(6-8*PO)
      CM3=1/(30-30*PO)
      CM4=(5*PO-1)/(15*(1-PO))
      CM5=2*PO/(3-6*PO)
      CM6=(4-5*PO)/(15*(1-PO))
      DO 300 I=1,NN
      DO 305 J=1,NN
      RM(1)=X(I)-X(J)
      RM(2)=Y(I)-Y(J)
      NORM(1)=NORM1(I)
      NORM(2)=NORM2(I)
      R2=SQRT(RM(1)**2+RM(2)**2)
      PM(I,J)=(RMAX-R2)
```

```
      DO 310 K=1,2
      DO 315 L=1,2
      IM=2*I+K-2
      JM=2*J+L-2
C     COMPUTE D MATRIX
      DM(IM,JM)=RO*((CM1*R2-CM2*RMAX)*D(K,L)*R2**2-
     *(CM3*RM(K)*RM(L)*R2))/GM
      IF(R2.LE..001)GO TO 330
C     COMPUTE T MATRIX
      TM(IM,JM)=RO*((CM4*R2-CM5*RMAX)*RM(L)*NORM(K)+(CM6*R2-2*CM2*RMAX)
     **RM(K)*NORM(L)+((CM6*R2-2*CM2*10.)*D(K,L)-(2*CM3*RM(K)*RM(L))/R2)
     **(RM(K)*NORM(K)+RM(L)*NORM(L)))
      GO TO 315
  330 TM(IM,JM)=RO*((CM4*R2-CM5*RMAX)*RM(L)*NORM(K)+(CM6*R2-2*CM2*RMAX)
     **RM(K)*NORM(L))
  315 CONTINUE
  310 CONTINUE
  305 CONTINUE
  300 CONTINUE
      RETURN
      END
C
C
C
C
C
C
C
C
      SUBROUTINE FUNC(ICOD,JA,C1,C2,C3,C4,C5,C6,C7,PO,II,IF,XS,YS,ISY,
     *IIS,IFS)
C     INTEGRALS OVER BOUNDRY ELEMENTS
      implicit real*8(A-H,O-Z)
      COMMON /A/ D(2,2),XI(6,3),W(6,3),IDUP(50),INC(50,2),C(50),
     *S(50,3),ISYM(100),X(100),Y(100),IFIP(100),AK(100,100),P(100),
     *XM(100),AG(100,100)
     *,PM(100,100),DM(100,100),TM(100,100)
      REAL*8 NORM1(100),NORM2(100),NORM(2)
      COMMON /A4/ H(3,4),G(3,4),HL(3,4),GL(3,4),NORM1,NORM2
     *,NORM,RM(2)
      DIMENSION DXY(2),BN(2),B(2),DR(2),UL(2,2),PL(2,2),ULL(2,2,2),
     *PLL(2,2,2)
      DO 5 KK=1,3
      DO 5 L=1,4
      GL(KK,L)=0.
      HL(KK,L)=0.
      G(KK,L)=0.
    5 H(KK,L)=0.
      DXY(1)=X(IF)-X(II)
      DXY(2)=Y(IF)-Y(II)
```

```
      BN(1)=DXY(2)/C(JA)
      BN(2)=-DXY(1)/C(JA)
      NORM1(IF)=BN(1)
      NORM2(IF)=BN(2)
      NORM1(II)=BN(1)
      NORM2(II)=BN(2)
      GO TO (1,2,2,1),ICOD
C     SELECT NO. OF INTEGRATION POINTS
      SEL=0.5*SQRT((2.*XS-X(II)-X(IF))**2+(2.*YS-Y(II)-Y(IF))**2)/C(JA)
    1 NPI=4
      IF(SEL.LE.1.5)NPI=6
      IF(SEL.GT.5.5)NPI=2
      INP=NPI/2
C     COMPUTE MATRICES NUMERICALLY
      DO 50 KK=1,NPI
      XMXI=0.5*(1.+XI(KK,INP))*DXY(1)+X(II)-XS
      YMYI=0.5*(1.+XI(KK,INP))*DXY(2)+Y(II)-YS
      R=SQRT(XMXI**2+YMYI**2)
      B(1)=-0.25*(XI(KK,INP)-1.)*C(JA)
      B(2)=0.25*(XI(KK,INP)+1.)*C(JA)
      DR(1)=XMXI/R
      DR(2)=YMYI/R
      DRDN=DR(1)*BN(1)+DR(2)*BN(2)
C     COMPUTE MATRICES H AND G
      DO 6 I=1,2
      DO 6 J=1,2
      UL(I,J)=-C1*(C2*DLOG(R)*D(I,J)-DR(I)*DR(J))
      PL(I,J)=-C3*((C4*D(I,J)+2.*DR(I)*DR(J))*DRDN+C4*(DR(J)*BN(I)-DR
     **BN(J)))/R
    6 CONTINUE
      DO 7 LA=1,2
      IC=0
      DO 7 LL=1,2
      DO 7 JJ=1,2
      IC=IC+1
      G(LA,IC)=G(LA,IC)+UL(LA,JJ)*B(LL)*W(KK,INP)
      H(LA,IC)=H(LA,IC)+PL(LA,JJ)*B(LL)*W(KK,INP)
    7 CONTINUE
C     COMPUTE MATRICES HL AND GL (INTERNAL STRESSES)
   10 DO 11 I=1,2
      DO 11 J=I,2
      DO 11 K=1,2
      ULL(I,J,K)=C3*(C4*(DR(J)*D(K,I)+DR(I)*D(K,J)-DR(K)*D(I,J)+2.*DR(I)
     *)*DR(J)*DR(K))/R
      B1=2.*DRDN*(C4*DR(K)*D(I,J)+PO*(DR(J)*D(I,K)+DR(I)*D(J,K)-4.*DR(I)
     *)*DR(J)*DR(K))
      B2=2.*PO*(BN(I)*DR(J)*DR(K)+BN(J)*DR(I)*DR(K))
      B3=C4*(2.*BN(K)*DR(I)*DR(J)+BN(J)*D(I,K)+BN(I)*D(J,K))
   11 PLL(I,J,K)=C6*(B1+B2+B3-C7*BN(K)*D(I,J))/R**2
      IL=0
```

```
      DO 12 I=1,2
      DO 12 J=I,2
      IL=IL+1
      IC=0
      DO 12 IAA=1,2
      DO 12 JAA=1,2
      IC=IC+1
      GL(IL,IC)=GL(IL,IC)+B(IAA)*ULL(I,J,JAA)*W(KK,INP)
   12 HL(IL,IC)=HL(IL,IC)+B(IAA)*PLL(I,J,JAA)*W(KK,INP)
   50 CONTINUE
      GO TO 18
C     COMPUTE MATRICES H AND G ANALYTICALLY (BOUNDRY CONSTRAINT EQ.)
    2 AL=C5*C2*C(JA)
      AA=AL*(0.5-DLOG(C(JA)))
      DO 15 I=1,2
      DO 15 J=1,4
      IT=(J/2)*2+2-J
      G(I,J)=C5*DXY(I)*DXY(IT)/C(JA)
      IF(IT.EQ.I)G(I,J)=G(I,J)+AA
   15 CONTINUE
      IAA=-2
      IF(ICOD.EQ.3)IAA=0
      G(1,3+IAA)=G(1,3+IAA)+AL
      G(2,4+IAA)=G(2,4+IAA)+AL
      H(1,2-IAA)=C3*C4*(1.+IAA)
      H(2,1-IAA)=-H(1,2-IAA)
C     SYMMETRY TEST
   18 IF(ISY.EQ.1)GO TO 8
      DO 24 I=IIS,IFS
      DO 24 J=1,4
      H(I,J)=-H(I,J)
   24 G(I,J)=-G(I,J)
      IF(ICOD.NE.4.OR.ISY.EQ.4)GO TO 8
      DO 25 J=1,4
      HL(2,J)=-HL(2,J)
   25 GL(2,J)=-GL(2,J)
    8 RETURN
      END
C
C
C
C
C
C     ADD TWO MATRICES
      SUBROUTINE MATADD(N,A1,A2,A3)
      implicit real*8(A-H,O-Z)
      REAL*8 A1(100,100),A2(100,100),A3(100,100)
      DO 10 I=1,N
      DO 10 J=1,N
      A3(I,J)=A1(I,J)+A2(I,J)
```

```
10      CONTINUE
        RETURN
        END
C
C
C
C       MULTIPLY TWO MATRICES
        SUBROUTINE MATMULT(N,M1,M2,M3)
        implicit real*8(A-H,O-Z)
        REAL*8 M1(100,100),M2(100,100),M3(100,100)
        DO 5 I=1,N
        DO 5 J=1,N
        M3(I,J)=0.
    5   CONTINUE
        DO 10 I=1,N
        DO 10 J=1,N
        DO 10 K=1,N
        M3(J,I)=M3(J,I)+M1(J,K)*M2(K,I)
10      CONTINUE
        RETURN
        END

C       MULTIPLY TWO COMPLEX MATRICES
        SUBROUTINE MATMULTI(N,M1,M2,M3)
        implicit real*8(A-H,O-Z)
        COMPLEX*16 M1(100,100),M2(100,100),M3(100,100)
        DO 5 I=1,N
        DO 5 J=1,N
        M3(I,J)=(0.D0,0.D0)
5       CONTINUE
        DO 10 I=1,N
        DO 10 J=1,N
        DO 10 K=1,N
        M3(J,I)=M3(J,I)+M1(J,K)*M2(K,I)
10      CONTINUE
        RETURN
        END
C
C
C
C
C       Formulates the [K] matrix of the particular integral method
        SUBROUTINE MATEXP(N,EX1,EX2)
        implicit real*8(A-H,O-Z)
        REAL*8 EX1(100,100),EX2(100,100),D(2,2)
        D(1,1)=1.
        D(2,2)=1.
        D(1,2)=0.
        D(2,1)=0.
        DO 10 I=1,N
```

```
      DO 10 J=1,N
      DO 10 K=1,2
      DO 10 L=1,2
      II=2*I+K-2
      JJ=2*J+L-2
      EX2(II,JJ)=EX1(I,J)*D(K,L)
   10 CONTINUE
      RETURN
      END
C
C
C
C     SUBTRACT TWO MATRICES
      SUBROUTINE  MATSUB(N,SUB1,SUB2,SUB3)
      implicit real*8(A-H,O-Z)
      DIMENSION SUB1(100,100),SUB2(100,100),SUB3(100,100)
      DO 10 I=1,N
      DO 10 J=1,N
      SUB3(I,J)=SUB1(I,J)-SUB2(I,J)
10    CONTINUE
      RETURN
      END
C
C
C
C
C
      SUBROUTINE FRESWP(NN2,NTL,M1,K1,G,RO,E,AREA,LENTH,INERT
     *,INC,XO,YO)
c     This subprogram multiplies the mass, [m] of Eq 33, matrix by a
c     given frequency, and
c     then forms the transfer
c     matrix. Once the transfer matrix is formed it is input into the  RGO
c     routines for eigenvalue and eigenvector extraction.
c
c     On input:
c
c     K1 is the matrix formulated in the boundary element calculations that
c     multiplies the displacement terms ( this is the H matrix )
c
c     M1 is the mass matrix calculated by the particular integral
c     technique.
c
c     G is the matrix formulated in the boundary element calculations that
c     multiplies the displacement terms.
c
c     RO and E are material properties. NN2 is the size of the above
c     matrices.
c     NTL is the number of nodes on the interface of the substructure.
c     XO,YO,INC are geometry descriptions used for post processing
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 M1(100,100),K1(100,100),G(100,100),GG(100,100)
     *,MM(100,100),IM(100,100),WKAREA(2600),M2(100,100),WR(100)
     *,WI(100),ZR(100,100),FV1(100),inert,lenth,IMAGX(100)
     *,ZI(100,100),IMAGY(100),REALX(100),REALY(100),G1(100,100)
     *,GN(100,100),EIMAG(100),PART5(100,100),PART6(100,100)
     *,FLR(50),FLI(50),FRR(50),FRI(50),XO(100),YO(100)
      INTEGER DMN,IV1(100),N,NTL4,INDEX(100),INC(50,2)
      OPEN(10,FILE='VECTOR.OUT',STATUS='NEW')
      DMN=100
      IDGT=0
      NTL4=4*NTL
C     this loop changes the frequency term if desired
      DO 100 N=1,1
      WRITE(9,110)
110   FORMAT(////)
      HERTZ= 100.
      fre=hertz*(2*3.141592654)
c
c     thphi is the theoretical value of the phase angle for compresional-
c     extentional motion of a beam.
c
      thphi=dsqrt(ro/e)*hertz*lenth*360
c
c     thphi2 is the theoretical value of the phase angle for bending of a
c     beam
c
      thphi2=dsqrt(fre*dsqrt(ro*area/(e*inert)))
      thphi2=thphi2*lenth*540/(2.5*3.141592654)
      DO 20 K=1,NN2
      DO 20 L=1,NN2
      M2(K,L)=(FRE**2)*M1(K,L)
20    CONTINUE
      CALL MATADD(NN2,K1,M2,MM)
      CALL INVERT(MM,M2,NN2,DMN)
      CALL MATMULT(NN2,M2,G,GG)
C-----------------------------------------------------------------------C---c

C      MATPART takes the dynamical admittance matrix G and manipulates it
c      into the substructure transfer matrix GG. This is done by
c      paritioning
c      the matrix, eliminating the interior nodes and rearanging terms in
c      order to seperate the left and right nodes.
c
      CALL MATPART(GG,NTL,PART5,PART6,NN2)
      matz=1
c*****************************************
c RGO calls the EISPACK routine for extracting the eigenvalues and
c eigenvectors of the transfer matrix.  The EISPACK routines are not
```

```
c included in this listing.
c*****************************************
c
      CALL RGO(DMN,NTL4,GG,WR,WI,MATZ,ZR,ZI,IV1,FV1,IERR)

c     The rest of the FRESWP is merely manipulation of the Eigenvalues and
c     eigenvectors for output.
      do 30 I=1,ntl4
      eimag(I)=dsqrt(wr(I)**2+wi(I)**2)
C      IF(TEST.NE.0.0) THEN
C      TEST=0.
C      GOTO 30
C      ENDIF
c
c     eimag is the magnitude of the eigenvector
c
c     phee is the phase angle
c
      phee=datan(wi(I)/wr(I))*(360.0/6.2831850)
      TEST = WI(I)
      TESIMAG=DABS(EIMAG(I)-1)
C      IF (EIMAG(I).gt.1.10D0) GOTO 30
C      IF(EIMAG(I).LT.1.D-4) GOTO 30
      WRITE(9,50)
50    FORMAT(/,2X'FREQUENCY(hertz)',5X,'COMPR PHASE(deg)',5x,
     *'BEND PHASE(deg)')
      WRITE(9,55)hertz,thphi,thphi2
55    FORMAT(E15.5,5X,E15.5,5X,E15.5,/)
      WRITE(9,60)
60    FORMAT(6X'REAL',13X,'IMAGINARY',7X,'EIMAG',10X,'PHASE')
      TEST=WI(I)
      WRITE(9,65)WR(I),WI(I),EIMAG(I),PHEE
65    FORMAT(E15.5,2X,E15.5,2X,E15.5,2X,E15.5,/)
      WRITE(9,70)
70    FORMAT(1X'NODE #',18X,'XDISP',31X,'YDISP')
      WRITE(9,75)
75    FORMAT(16X,'REAL',10X,'IMAG',16x,'REAL',16X,'IMAG')
      DO 15 J=1,ntl
      IX=J*2-1
      IY=J*2
      IMAGX(j)=ZI(IX,I)
      IMAGY(j)=ZI(IY,I)
      REALX(J)=ZR(IX,I)
      REALY(J)=ZR(IY,I)
10    WRITE(9,40)J,ZR(IX,I),IMAGX(J),ZR(IY,I),IMAGY(J)
40    format(I5,4X,E15.5,2x,E15.5,4x,E15.5,1x,E15.5)
15    continue
      DO 150 J=1,2*NTL
      K=2*NTL+J
      FLR(J)=ZR(K,I)
```

```
      FLI(J)=ZI(K,I)
      FRR(J)=ZR(K,I)*WR(I)-ZI(K,I)*WI(I)
      FRI(J)=ZR(K,I)*WI(I)+ZI(K,I)*WR(I)
150   CONTINUE
      DO 160 K=1,NN2/2-2*NTL
      KI=NTL*2+K
      IX=2*K-1
      IY=2*K
      IMAGX(KI)=0.D0
      IMAGY(KI)=0.D0
      REALX(KI)=0.D0
      REALY(KI)=0.D0
      DO 170 J=1,2*NTL
      IMAGX(KI)=-FLI(J)*PART5(IX,J)+FRI(J)*PART6(IX,J)+IMAGX(KI)
      IMAGY(KI)=-FLI(J)*PART5(IY,J)+FRI(J)*PART6(IY,J)+IMAGY(KI)
      REALX(KI)=-FLR(J)*PART5(IX,J)+FRR(J)*PART6(IX,J)+REALX(KI)
      REALY(KI)=-FLR(J)*PART5(IY,J)+FRR(J)*PART6(IY,J)+REALY(KI)
170   CONTINUE
160   CONTINUE
c     Vector is a user supplied routine for doing structure plots
C      CALL VECTOR (NTL,REALX,REALY,IMAGX,IMAGY,WR(I),WI(I),INC
C     *,XO,YO,NN2/2)
30    continue
100   CONTINUE
      return
      end
C
C
C
C
C
      SUBROUTINE MATPART(GG,NTL,PART5,PART6,NN2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PART1(100,100),PART3(100,100),PART4(100,100),
     *A(100,100),B(100,100),C(100,100),D(100,100),
     *BINV(100,100),GG(100,100),PART5(100,100),PART6(100,100)
      INTEGER DMN
      DMN=100
      DO 10 I=1,2*NTL
      DO 10 J=1,2*NTL
      A(I,J)=GG(I,J)
      k=2*ntl+j
      B(I,J)=GG(I,k)
      l=2*ntl+i
      C(I,J)=GG(l,J)
      D(I,J)=GG(l,k)
10    CONTINUE
      DO 50 I=1,NN2-4*NTL
      K=4*NTL+I
      DO 50 J=1,2*NTL
```

```
      JI=2*NTL+J
      PART5(I,J)=GG(K,J)
      PART6(I,J)=GG(K,JI)
50    CONTINUE
      ntl2=2*ntl
      CALL INVERT(B,BINV,NTL2,DMN)
      CALL MATMULT(NTL2,BINV,A,PART4)
      CALL MATMULT(NTL2,D,BINV,PART1)
      CALL MATMULT(NTL2,D,PART4,PART3)
      CALL MATSUB(NTL2,PART3,C,PART3)
      DO 30 I=1,2*NTL
      DO 30 J=1,2*NTL
      GG(I,J)=PART1(I,J)
      k=2*ntl+j
      l=2*ntl+i
      GG(I,k)=PART3(I,J)
      GG(l,J)=BINV(I,J)
      GG(l,k)=PART4(I,J)
30    CONTINUE
      RETURN
      END
c
c
c
c
      SUBROUTINE INVERT(A,Y,N,NP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(100,100),Y(100,100),INDX(100)
      DO 12 I=1,N
         DO 11 J=1,N
         Y(I,J)=0.
11       CONTINUE
      Y(I,I)=1.
12    CONTINUE
      CALL LUDCMP(A,N,NP,INDX,D)
      DO 13 J=1,N
      CALL LUBKSB(A,N,NP,INDX,Y(1,J))
13    CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE LUDCMP(A,N,NP,INDX,D)
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER (NMAX=100,TINY=1.0E-20)
      DIMENSION A(100,100),INDX(100),VV(100)
      D=1.
      DO 12 I=1,N
      AAMAX=0.
```

```
      DO 11 J=1,N
      IF (DABS(A(I,J)).GT.AAMAX) AAMAX=DABS(A(I,J))
11    CONTINUE
      IF (AAMAX.EQ.0.)  PAUSE 'SINGULAR MATRIX'
      VV(I)=1./AAMAX
12    CONTINUE
      DO 19 J=1,N
      DO 14 I=1,J-1
      SUM=A(I,J)
      DO 13 K=1,I-1
      SUM=SUM-A(I,K)*A(K,J)
13    CONTINUE
      A(I,J)=SUM
14    CONTINUE
      AAMAX=0.
      DO 16 I=J,N
      SUM=A(I,J)
      DO 15 K=1,J-1
      SUM=SUM-A(I,K)*A(K,J)
15    CONTINUE
      A(I,J)=SUM
      DUM=VV(I)*DABS(SUM)
      IF(DUM.GE.AAMAX) THEN
      IMAX=I
      AAMAX=DUM
      ENDIF
16    CONTINUE
      IF(J.NE.IMAX) THEN
      DO 17 K=1,N
      DUM=A(IMAX,K)
      A(IMAX,K)=A(J,K)
      A(J,K)=DUM
17    CONTINUE
      D=-D
      VV(IMAX)=VV(J)
      ENDIF
      INDX(J)=IMAX
      IF(A(J,J).EQ.0.) A(J,J)=TINY
      IF(J.NE.N) THEN
      DUM=1./A(J,J)
      DO 18 I=J+1,N
      A(I,J)=A(I,J)*DUM
18    CONTINUE
      ENDIF
19    CONTINUE
      RETURN
      END
C
C
C
```

```
C
      SUBROUTINE LUBKSB(A,N,NP,INDX,B)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(100,100),INDX(100),B(100)
      II=0
      DO 12 I=1,N
      LL=INDX(I)
      SUM=B(LL)
      B(LL)=B(I)
      IF (II.NE.0) THEN
      DO 11 J=II,I-1
      SUM=SUM-A(I,J)*B(J)
11    CONTINUE
      ELSEIF (SUM.NE.0.) THEN
      II=I
      ENDIF
      B(I)=SUM
12    CONTINUE
      DO 14 I=N,1,-1
      SUM=B(I)
      IF(I.LT.N) THEN
      DO 13 J=I+1,N
      SUM=SUM-A(I,J)*B(J)
13    CONTINUE
      ENDIF
      B(I)=SUM/A(I,I)
14    CONTINUE
      RETURN
      END
```

Bibliography

1. Cremer L., and Lielich H.O. "Zur Theorie der Biegekettenleiter," Archiv der Elektrischhen Ubertrag, 7, 1953.

2. Heckel M.A. "Investigations on the Vibrations of Grillages and Other Simple Beam Structures," Journal of the Acoustical Society Of America, 36, 1964.

3. von Flotow A.H. "A Traveling Wave Approach To the Dynamic Analysis of Large Space Structures," AIAA Paper 83-0964, 1983.

4. Signorelli, J. Wave Propagation in Periodic Truss Structures, MS Thesis, Massachusetts Institute of Technology, Feb 1987.

5. Saada, A.S., Elasticity: Theory and Applications, Florida: Robert E. Kreiger Publishing Company, 1974.

6. Danson, D.J. "Elasticity Problems," Boundary Element Techniques in Computer - Aided Engineering, Edited by: Brebbia, C.A., Dordrect: Martinus Nijhoff Publishers, 1984.

7. Gibson B.S., Boundary Element Fundamentals: Basic Concepts and Recent Developments in the Poisson Equation, (Volume 2) Southhampton UK and Boston USA: Computational Mechanics Publications, 1987.

8. Ahmad, .S, and Banerjee, P.K., "Free Vibration Analysis by BEM Using Particular Integrals," Journal of Engineering Mechanics, Vol 112, No. 7, July 1986.

9. Brebbia, C.A., Telles, J.C.F., and Wrobel, L.C., Boundary Element Techniques, Berlin: Springer-Verlag, 1984.

10. Smith B.T. et al. Matrix Eigensystem Routines (Lecture Notes in Computer Science; 6), (Second Edition), Berlin: Springer Verlag, 1976.

11. Meirovitch, L. Analytical Methods in Vibrations, New York: Macmillan Publishing Co;,Inc., 1967.

Vita

Captain Matthew C. Phillips [redacted]. He grew up in Portland, Oregon where he graduated from David Douglas High School in 1978. On 20 April 1984 he received a B.S. in Mechanical Engineering from Brigham Young University. He received his Air Force commission on 7 August 1984 as a Distinguished Graduate of The United States Air Force Officer Training School. After commissioning, he worked at The Air Force Weapons Laboratory at Kirtland AFB, New Mexico as a structural engineer in support of the SDI Large Space Structures Program. In May 1987 he enrolled in the Air Force Institute of Technology to pursue a M.S. Degree in Aeronautical Engineering.

He married the former Miss Lisa Dawn Pratt of Portland, Oregon in 1979, They presently have four children.

[redacted]

UNCLASSIFIED
ECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GAE/ENY/89D-28	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (*If applicable*)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (*City, State, and ZIP Code*): Air Force Institue of Technology, Wright Patterson AFB OH 45433-6583	7b. ADDRESS (*City, State, and ZIP Code*)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (*If applicable*)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (*Include Security Classification*)

DYNAMIC STRUCTURAL ANALYSES USING BOUNDARY ELEMENT TECHNIQUES

12. PERSONAL AUTHOR(S)
Matthew C. Phillips, B.S., Capt, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (*Year, Month, Day*)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	1990 May	72

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	Boundary Elements Wave Propagation Dynamic Structural Analyses BEM

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

Thesis Advisor: Lanson J. Hudson, Maj, USAF
AFWAL/FDSRC
Wright Patterson AFB OH

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (*Include Area Code*)	22c. OFFICE SYMBOL
Lanson J. Hudson, Maj, USAF	AV 785-7384	AFWAL/FDSRC

DD Form 1473, JUN 86 *Previous editions are obsolete*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED